Generating the Animation of a 3D Agent from Explanation Text

Ken'ichi KAKIZAKI
Department of Computer Science and Electronics
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka, 820-8502, Japan.
kakizaki@cse.kyutech.ac.jp

1. ABSTRACT

This paper proposes a presentation agent designed for a virtual environment. The agent, animated in the virtual environment, performs a presentation based on a speech text that explains objects in the environment. In order to generate the agent's motion automatically, we categorize the presentation motions of the agent into three classes, pointing, moving, and gesturing. Based on this categorization, our system extracts words from the explanation text. In order to determine what the target object is and to extract detailed information about the object, the system accesses a scene graph that contains all the information about the virtual environments. The system can find the object based on its characteristics that are explained in the text, and can then extract information about the object such as its size, position, and weight. This information is used for determining details to generate the agent's motion.

1.1 Keywords

Presentation agent, synthetic agent, computer animation, animation authoring, virtual environment, motion generation, scene graph.

ACM Multimedia'98, Bristol, UK
© 1998 ACM 1-58113-036-8/98/0008 $5.00

2. INTRODUCTION

In recent years, implementation methods of distributed virtual environment have been researched, and the technology is expected to become an excellent service platform in the network. In addition, software agents are indispensable to improve the service quality in these distributed virtual environments. Some agents in a virtual environment [1, 2, 3] are already proposed. In this paper, we introduce a presentation agent in a distributed virtual environment. The agent performs a presentation based on an explanation text.

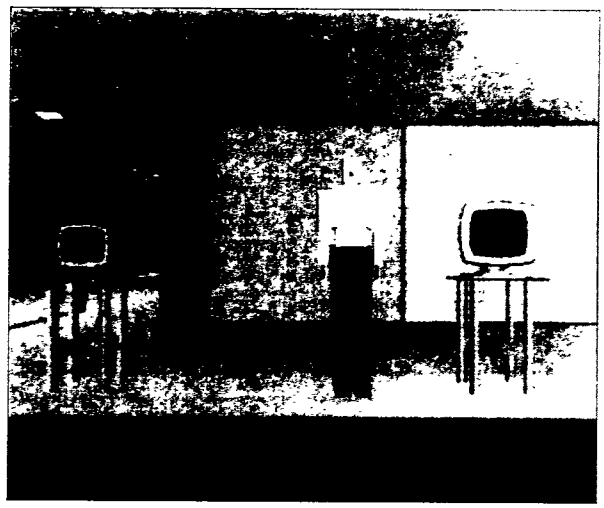


Figure 1: Screen shot of a presentation.

Presentations are very important activities in academic, educational, and commercial settings. Many presentations are often performed in many different places. However, these kinds of ordinary presentations have several limitations. For example, audiences wishing to attend a presentation must go to where the presentation is being held. A presentation agent in a distributed virtual environment, however, can resolve these problems easily. A screen shot of the service is shown in Figure 1. Thus, an audience can access a service presented in a virtual environment setting from anywhere and at anytime.

The use of a presentation agent has many advantages. However, the motion authoring and programming of the agent have to be done as a 3D basis [4], so it is often very difficult for people less familiar with these procedures. In order to avoid difficulties, an automatic motion generating method for agents [5, 6] is introduced. The input information of the system is an explicit description of the motion. This method makes the system generating motions easy, but the author still has to describe the motion clearly. This work is also not so easy for those less technically oriented. Therefore, the development of easy authoring method is strongly required to make the presentation method popular.

The method we introduce in this paper realizes easy authoring for the presentation agent. The input information in our method is in the form of an explanation text that the agent will speak. This kind of explanation speech text is usually written by the people who want to make a presentation. Therefore, it has the advantage that any individual can prepare his or her own presentation text without too much trouble. The motions performed by the agent, however, are not expressed explicitly in the text, which means our system has to guess the motion. Our system executes a guess based on the explanation text and the scene graph which contains information about all of the objects in the virtual environment.

3. PRESENTATION IN THE VIRTUAL ENVIRONMENT

As already mentioned, conventional presentations are usually performed in person, but such presentations have many limitations. A presentation in a virtual environment by an agent, on the other hand, has many advantages. In this section, we discuss the advantages of the presentation agent in the distributed virtual environment.

3.1 Time and Space

Live presentations performed by persons are the standard way to present. However, these presentations have a drawback in that both audiences and presenters must meet at specific location and at specific time. Such a limitation can create problems for both parties. Presenters usually spend much in the way of time, energy, and money towards a presentation, which leads the teachers to want to show the presentation to many audiences. Similarly, if audiences want to see a particular presentation, they may have difficulty attending it if the presentation's location and schedule does not match their own.

3.2 Viewpoint

If we want to avoid some of the limitations of a live presentation, we might use recorded media such as videos and slide-shows. However, such a presentation is recorded as a 2D basis. In such media, the viewpoint is fixed at the time of creation and the audience can't move their viewpoint to where they want. A presentation in the virtual environment, on the other hand, is performed as a 3D basis. This gives students the opportunity to shift their viewpoint to any location they chose.

3.3 Environment

In ordinary presentations, the variety of visual aides of the presentation environment is restricted by many practical elements such as money, time, space, and so on. However, in a presentation in a virtual environment, the author is able to prepare any kind of environment he or she can imagine. This method is especially helpful if an object that the presenter wants to explain does not exist in the world, the virtual environment then becomes an ideal environment to do this presentation.

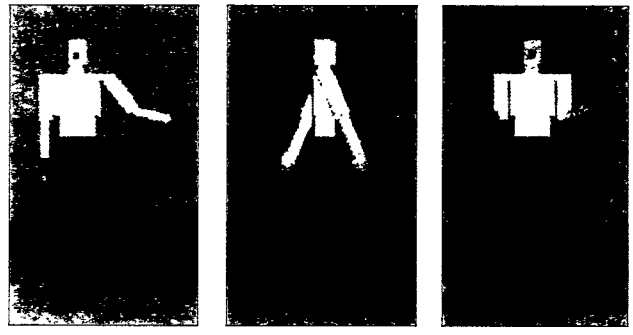
3.4 Repetition

It is usually required that a presentation be performed many times. However, performing an ordinal presentation many times is difficult, because additional expenditures in effort and cost are required for each presentation. In virtual environment presentations on the other hand, such additional effort and cost are not required. The agent just replays the presentation when necessary.

4. CLASSIFICATION OF MOTION

In order to generate the motions of an agent automatically, we have to know what the agent's motion is in a presentation and when the motion is performed. We believe that the primary objective of presentations is to show target objects clearly, then to talk about their features. Based on the objective, we classify the essential motions of an agent into three categories, as follows.

- Pointing
- Moving
- Gesturing



1: Pointing 2: Moving 3: Gesturing

Figure 2: Motion Categories.

We understand that presentation behavior is constructed from these motions, shown in Figure 2. In this section, we introduce the motions and discuss how can we guess the motion from the explanation text, and what kind of information is required for the motion generation.

4.1 Pointing

Pointing is a motion that the agent uses to indicate a target object. In a presentation, we usually point to an object that we are referring to, and it is this pointing motion which is used to identify that specific object for the audience. We believe the agent should perform a pointing motion when the target object is referred to in the explanation text.

The motion is basically performed by the agent's index finger, but the pointing motions have some versatility for example, an agent can:

- Point to an object with its index finger.
- Point to an object with an opened hand.
- Hold the object with a hand.

These pointing motions are performed in different ways depending on the situation. For example, if the target object is small, light weight, and near the agent, the agent could hold the target object in its hand. The decision table for the pointing method is shown in Table 1.

Table 1: Decision table for pointing method.

The number of object(s)	Position	Weight and size	Pointing method
Single	Far	-	Index finger
	Near	Light and small	Hold by hand
		Heavy or large	Point to with opened hand
Multiple	-	-	Point to with opened hand

In order to generate the pointing motion, the system basically has to know which object the agent should point to. In addition, the system also has to know the direction and the distance of the object from the agent. Moreover, in order to make this variety of pointing motions, the system has to know the size and weight of the object. The system extracts this kind of information from the scene graph of the virtual reality system, which will be described later.

4.2 Moving

Moving is a motion that the agent does to move from one position to another. Moving is usually performed by the agent to locate itself near the target object that will be explained. In the explanation text, the requirement of a moving motion is not explicitly expressed. However, the system can guess the motion from the explanation text, as to when and where the agent has to move.

In the case of the explained object being near the agent, the agent does not have to move. The agent just points to it and continues the explanation. In a different case, in which the explanation text refers to a distant object, demonstrative adjectives such as, "this" and "these" are used; the agent has to move near the object before the explanation will start.

In order to generate the moving motion, the system basically has to know what the object is that the text is currently explaining. In addition, the system also has to know the direction and the distance of the object in relation to the agent. The system also extracts this kind of information from the scene graph of the virtual reality system.

4.3 Gesturing

Gesturing is a motion that we use a great deal in live presentations, and it is a motion that we use to increase the power of expression in a presentation. Gestures are usually performed in conjunction with words such as adjectives, adverbs, interjections, and verbs. Therefore, the agent might perform such appropriate gestures when these types of words appear in the explanation text. We show some examples of gesture below.

- Counting on fingers.
- Explaining size or length with fingers and hands.
- Explaining shape with fingers or hands.
- Explaining motion with fingers or hands.

In this method, gesturing motions are decided upon only from an explanation text, so gestures that are associated with a particular word always appear with those words. However, we sometimes use different gestures for the same words. For example, in the case that we talk about something large, if an object's size is really large, we will open our arms to explain the size. If the typical size of an object is very small, but the object we explain is relatively large, we will open our index finger and thumb to explain the size. In order to create these gestures, the system has to know the typical size of the target object. The system also extracts this kind of information from the scene graph of the virtual reality system.

5. GENERATING PRESENTATION

In this chapter, we describe a motion generation method for a presentation agent.

5.1 Explanation Text Parsing

In our method, primary motions of the agent are generated from information about the explained object. Therefore, the system has to decide what the explained object is. The process is realized by the system by picking up a noun

clause which mentions a target object in the explanation text. Therefore, our system doesn't understand the meaning of an explanation text, rather the system just picks up a word or words.

Sometimes, a target object is referred by a pronoun. In this case, the system makes an anaphoric reference and refers to the previous noun as the description of the target object. The explanation text is constructed from well-developed and well-formed sentences. In the sentences, a target object is described clearly, so picking up a noun clause is usually not difficult.

The text parsing system also picks up words, such as verbs and interjections, to generate appropriate gesturing.

5.2 Scene Graph as Database

As described in chapter 4, each class of motion needs not only the information from the explanation text, but also any extra information to generate the motion appropriately. In the real world, determining a target object from its specific features and getting some information related to that object is very difficult for a computer, because the computer usually does not have the information about objects in the world. In virtual reality systems on the other hand, information about the objects in the environment is stored in a database called the scene graph [7, 8, 9]. Therefore, it is possible to determine a target object from its specific features and to get some information related to that object.

The scene graph was originally designed for real-time 3D graphics system. For example, VRML (Virtual Reality Modeling Language) is widely used in Internet, and it also uses scene graphs as the scene description. The scene graph contains data about the objects in a scene, such their shape, size, color, position. Each piece of information is stored as a node in the scene graph. In the scene graph, these nodes are arranged hierarchically in the form of a tree structure. The 3D graphics system traverses the tree structure to get information, and renders appropriate scenes based on the information.

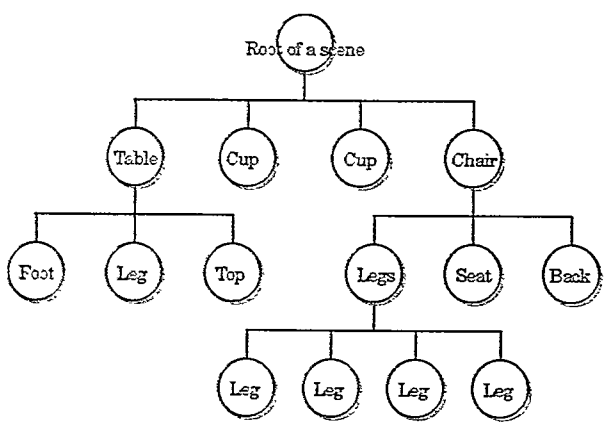


Figure 3: An Example of Scene Graph.

An example of a scene graph is shown in Figure 3. This scene graph describes a virtual environment that has four objects, or one table, one chair, and two cups, as shown in Figure 4. All of the information needed to render this environment is stored in the scene graph. In the scene graph, the major characteristics of four objects are described by four nodes right under the root node. The table and the chair have their sub-nodes, and these nodes indicate the visible structure and detailed information of these objects hierarchically.

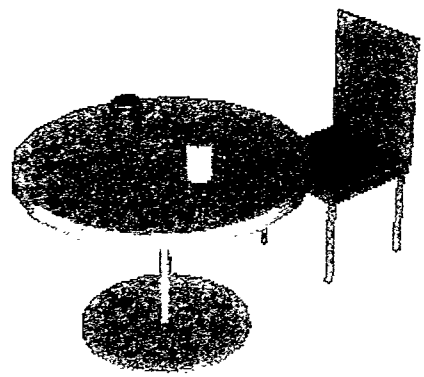


Figure 4: An Example of Virtual Environment.

The concept of the scene graph is very popular for constructing virtual reality systems. In the virtual reality system, the scene graph is extended for simulation. The scene graph for the virtual reality system contains not only data for rendering, but also contains data for simulation and human interface such as weight, and name. Therefore, the scene graph for a virtual reality system would have the following kinds of information about objects:

- Shape
- Size
- Color
- Position
- Weight
- Name

If we know some characteristics about an object, we are able to look up the object using this existing information in the scene graph. The system extracts these characteristics from the explanation text, and following that, the system traverses the scene graph to look up nodes that have appropriate characteristic. In VRML scene graph, for example, shape and size of an object is described in a geometry node, color of an object is described in an appearance node, and position of an object is described in a transform node. If the node is found, an object that contains the node in the sub-tree becomes a candidate of the target

object. After fixing the target object, the system is able to extract additional information from the scene graph to generate the agent's motions.

5.3 Motion Generation

The target object is basically referred to by a noun in an explanation text, as the following example will show:

I will talk about a cup.

In this case, the system picks up the noun "cup" and looks for the object by using the scene graph database, for instance shown in Figure 3. The cup is found in the scene graph, then the system knows its position. Based on the position, the system calculates its direction and distance from the agent. If the object is near the agent, the system generates a pointing motion in the agent. If the distance is far from the agent, on the other hand, the system additionally generates moving motion in the agent, preceding the pointing motion.

If many objects are listed, the system chooses the nearest object to the agent. This system decision is based on the heuristics that presenters usually present the object nearest to them. Sometimes, this heuristics chooses a different object from the target object that an author assumes. In this case, the author should add a description of the feature for the particular object into the explanation text to specify the target object exactly.

The target object can be referred to by a noun with adjectives that specify the features of the object, as shown below.

I will talk about a red cup.

In this case, the system also picks up the noun "cup" and looks for the object by the scene graph database. As a result, several objects are listed, and the adjective information "red" is also used to determine which object is the appropriate one. Consequently, the system generates the motion of the agent.

We can explain characteristics of a target object, also in this way, as shown below.

This cup is large.

In this case, the system, first, make a list of all cups existing in the virtual environment. Next, the system calculates the average size of these cups. After this processing, the system generates an appropriate degree of gesture to explain the characteristics.

5.4 Motion Scheduling

The motions that the agent should perform are generated according to an explanation text. Therefore, the order of the motion sequence is defined by the explanation text, and this sequence is performed sequentially. However, the raw sequence of motions has some problems, so a motion scheduling phase is required.

The main purpose of motion scheduling is shown below:

(1) To synchronize motions with the speech

(2) To permit motion overlapping

Presentations in virtual environments are performed with speech and motions. It is imperative that these motions synchronize with the speech. For example, if an agent talks about an object, the agent should already be pointing at this target object. In order for this to occur, all motions performed by the agent must start in advance of the pointing reference. These kinds of adjustments are performed by the scheduler.

In order to synchronize motions with the speech, another kind of adjustment is also important. The duration time of the motion is extracted from two sets of data. One is the typical execution time of each motion, and the other is the pronunciation time in the explanation text. In our system, management of duration time is performed in each sentence. If the total duration time of motions is longer than that of pronunciation, silent time of speech occurs. In a presentation, silent times of speech are often awkward and are undesirable. Therefore, the scheduler tries to reduce these silent times.

To do this, the scheduler tries to overlap some motions in a sentence to reduce a silent duration. Pointing and moving motions can always overlap, because pointing motions use hands and arms, and moving motions use legs, so both motions are able to be performed at the same time.

On the other hand, gesturing and some of the other motions sometimes cannot overlap. We believe that the importance of gesturing is lower than that of pointing and moving. Therefore, if the motion of gesturing conflicts with the motion of pointing and moving, the gesturing motion is not performed.

6. IMPLEMENTATION

We have developed an experimental system on the SGI's Cosmo Player. Cosmo Player is a VRML browser, and it can be programmed with Java language. VRML and Java are platform independent technologies for the WWW, so the presentation on the system is able to be distributed anywhere in the world.

The system is constructed from two parts, one is a scene manager that shows the virtual environment, and the other is a presentation generator that generates motions for a presentation agent. We show the system overview in Figure 5. Each part of the system is constructed from some sub-systems.

The presentation generator includes two databases, a word dictionary, and a gesture library. The word dictionary stores words and these parts of speech, such as "noun" and "verb". The information in the dictionary is used for a text parsing. The gesture library stores sequences for gesture motions associated with words. These motions are previ-

ously arranged and stored as generic motion sequences, instead of for a particular explanation text.

The scene manager is constructed from a rendering system and its management logic. Cosmo Player is used as the rendering system, and the management logic written in Java is connected with the Cosmo Player using EAI (External Authoring Interface).

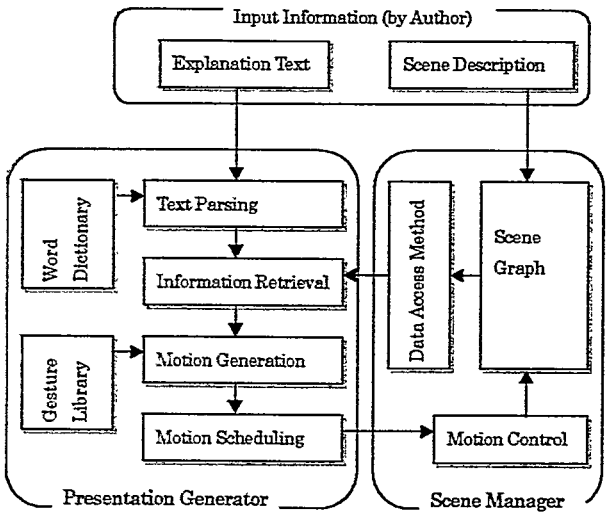


Figure 5: System Overview.

7. RESULT AND FUTURE WORK

We have confirmed that our experimental system generates presentation motions from just an explanation text and a scene graph. We believe that the generated motions for an agent were interesting enough as a presentation. However, we got some subjects to work out the solutions from the evaluation results.

Motion Library: The most conspicuous problem of our system that many audiences pointed out is the awkward motions of the presentation agent. The motion sequences used in this system are written by our hands numerically, because we don't have any CAD or motion capture system to develop the motion library. Therefore, none of motions, walking, pointing, and gesturing are well designed. For creating impressive presentation, we must refine the motion library and improve motion generation routines. To do the work, we will construct tools for motion library creation.

Text Parsing: The function of the current text parsing system is too simple to retrieve sufficient information form explanation text. Therefore, the motions that current system generates are relatively few and sometimes mismatched with the meaning of a text. In order to increase generating motions, and to reduce the mismatched motions, we should improve the function of text parsing system. Especially, we believe that a context management facility for an explanation text is important, and we will add the feature to our system.

Scene Graph Access Method: We believe that the scene graph access method for information retrieval is one of the most interesting areas in our research. We have confirmed that the basic functionality for a scene graph as a database. However, current system can handle only simple search requests. We have a plan to improve the search functionality for the scene graph. This technology will play an important role for the virtual reality and man-machine interface related applications.

8. CONCLUSION

In this paper, we introduced an automatic motion generating method for a presentation agent in a virtual environment. It generates an agent's motion from an explanation text and from information that is held in a scene graph. Our method makes it possible for those without a technical background to create individualized presentations in a virtual environment.

9. REFERENCES

[1] Rickel, J., Johnson, W., L.: "Integrating Pedagogical Capabilities in a Virtual Environment Agent", In Proceedings of Agents'97, (1997)

[2] Cassel. J., et al.: "Animated Conversation: Rule-based Generation of Facial Expression, Gesture & Spoken Intonation for Multiple Conversational Agents", In Proceedings of SIGGRAPH '94, (1994).

[3] Geib, C., Levison, L., Moore, M., B.: "SodaJack: An Architecture for Agents that Search for and Manipulate objects", In Proceedings of AAAI '94, (1994).

[4] Vince, J.: "3-D Computer Animation", Addison Wesley, (1992).

[5] Perlin, K., Goldberg, A.: "Improv: A System for Scripting Interactive Actors in Virtual Worlds", In Proceedings of SIGGRAPH '96, (1996).

[6] Wavish, P., Connah, D.: "Virtual actors that can perform scripts and improvise role", In Proceedings of Agents'97, (1997).

[7] Hartman, J., Wernecke, J.: "The VRML 2.0 Handbook", Addison Wesley, (1996).

[8] Wernecke, J.: "The Inventor Mentor", Addison Wesley, (1994).

[9] Sense8: "WorldToolKit Reference Manual, Release 6", (1996).